- $\hat{y}_n = +1$, $y_n = -1$ => false positive (FP) mistake.
- $\hat{y}_n = -1$, $y_n = +1$ => false negative (FN) mistake.

- Q: Will the updated w do "better" on the example x_n that it previously misclassified?

Vector difference $w - x \longrightarrow w$

~~for i in range(w.shape[0]):~~
~~w[i] = w[i] - x[i]~~

$\Longleftrightarrow$ vectorized version

$w = w - x$

Normally, training set is shuffled between epochs.

Perceptron update rule: if $\underbrace{h_n}_{\text{(}z_n\text{ logit score)}}$ $w^T x_n \cdot t_n \leq 0$

$$w = w + \eta t_n x_n$$

Q: Let $w^{(1)}$ be the params. obtained when $\eta = 1$

$w^{(2)}$ — " — " — when $\eta = 2$. > Train Perceptron to get these.

Let x be some arbitrary example.

$w^{(2)} x = w^{(1)} x$

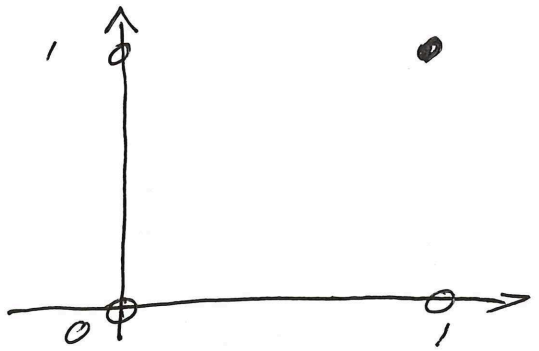
if $w^{(1)} x \geq 0 \Rightarrow w^{(2)} x \geq 0$

$w^{(1)} x < 0 \Rightarrow w^{(2)} x < 0$.

$$X = \begin{bmatrix} 0, 0 \\ 0, 1 \\ 1, 0 \\ 1, 1 \end{bmatrix}$$

$$y = [-1, -1, -1, +1]$$

add bias feature to each feature vector.



use column_stack in np.

$$\begin{bmatrix} 1, & 0, 0 \\ 1, & 0, 1 \\ 1, & 1, 0 \\ 1, & 1, 1 \end{bmatrix}$$

w_0 w_1 w_2

```
def train(X, y):
    w = 0
    .
    .
    .
    return w
```